Managing User Privileges and Operations with Authorization Services

Session 505

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Managing User Privileges and Operations with Authorization Services

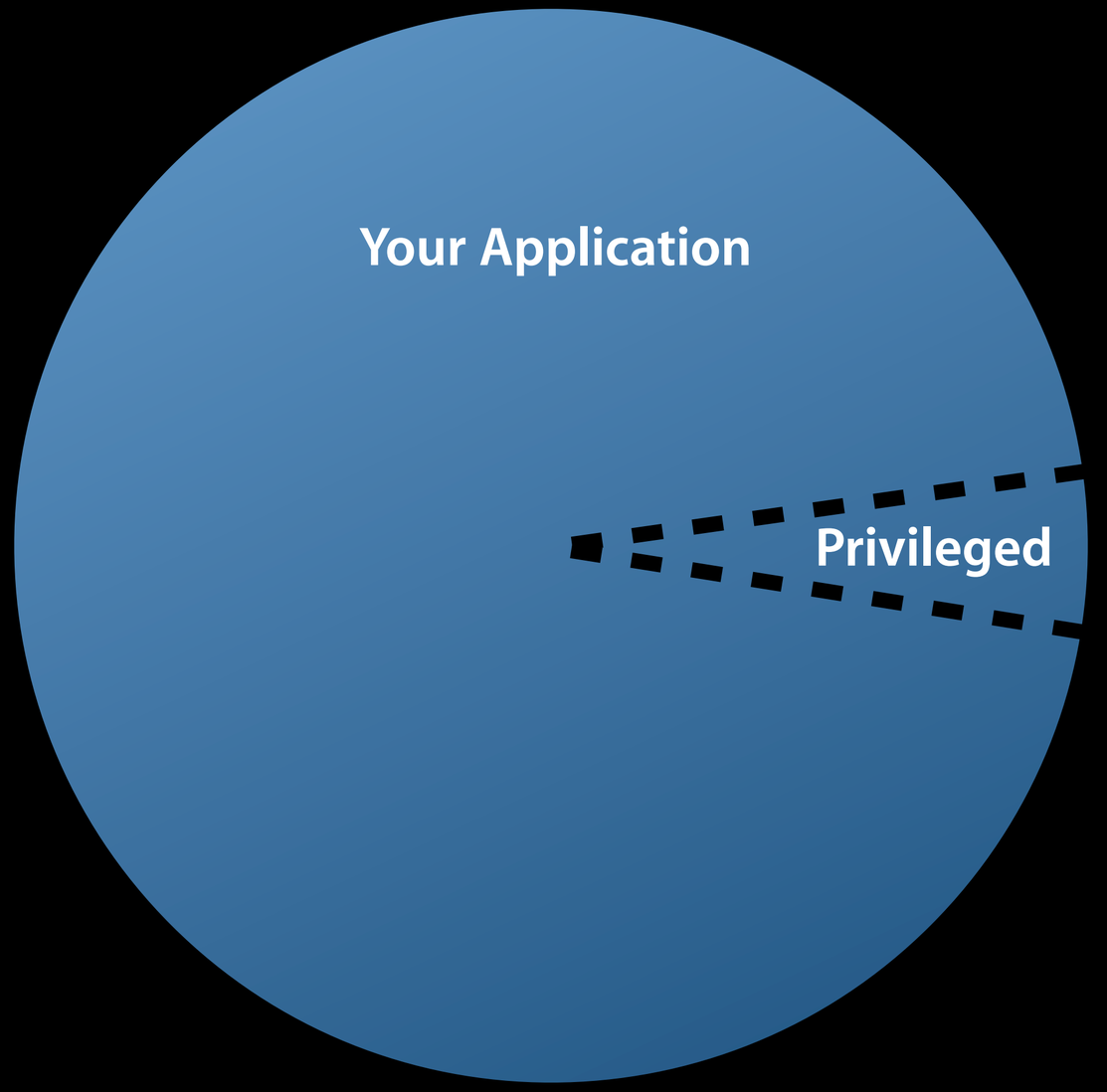
Disco Vince Giffin
OS Security Engineer

Least Privilege

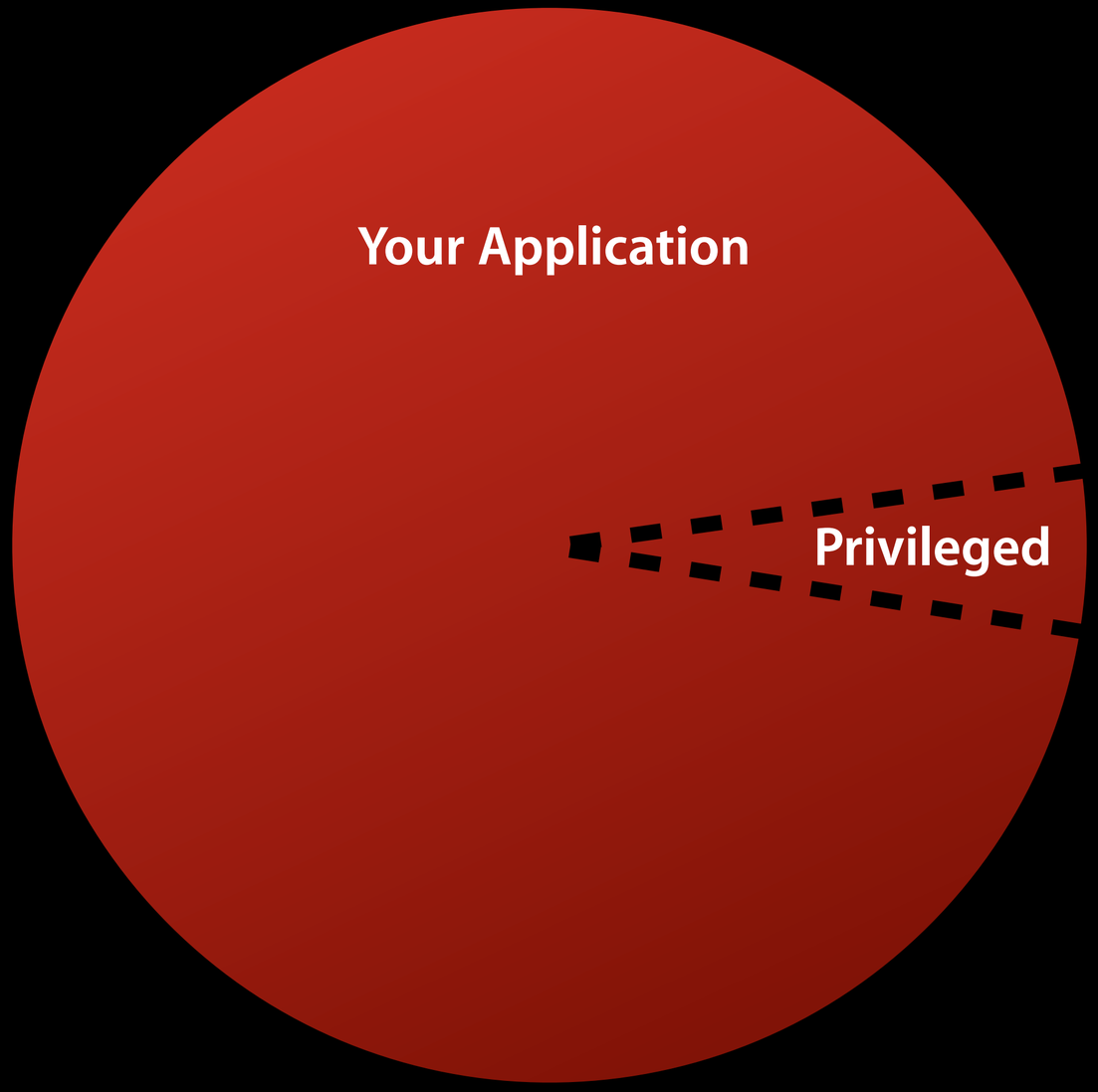
- Privileged operations
- *Something special*
  - Change the system time
  - Load or unload a kext
  - Modify a restricted part of the filesystem
- Perform them securely

First, let’s look at why this is so important.

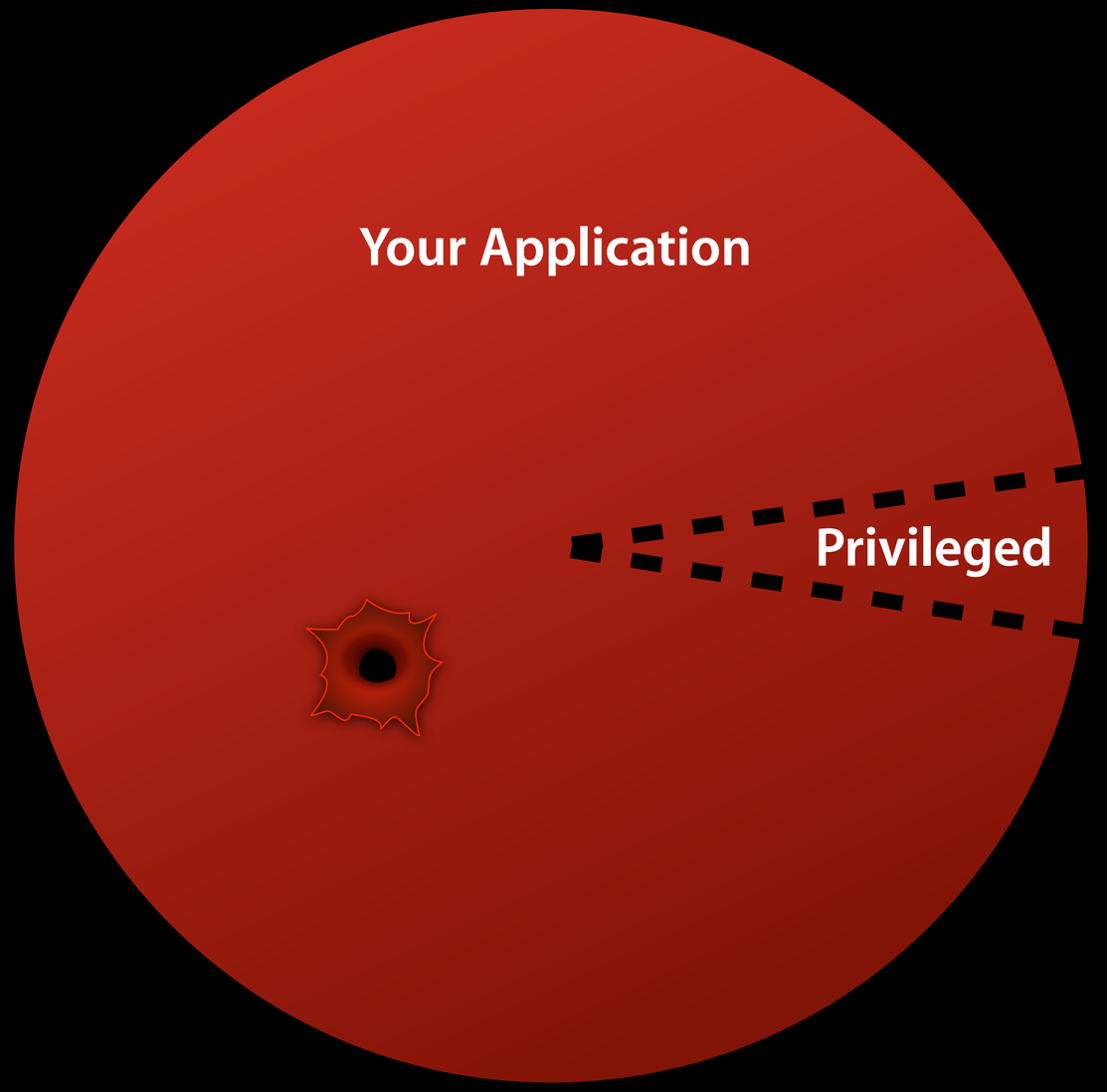
Least Privilege



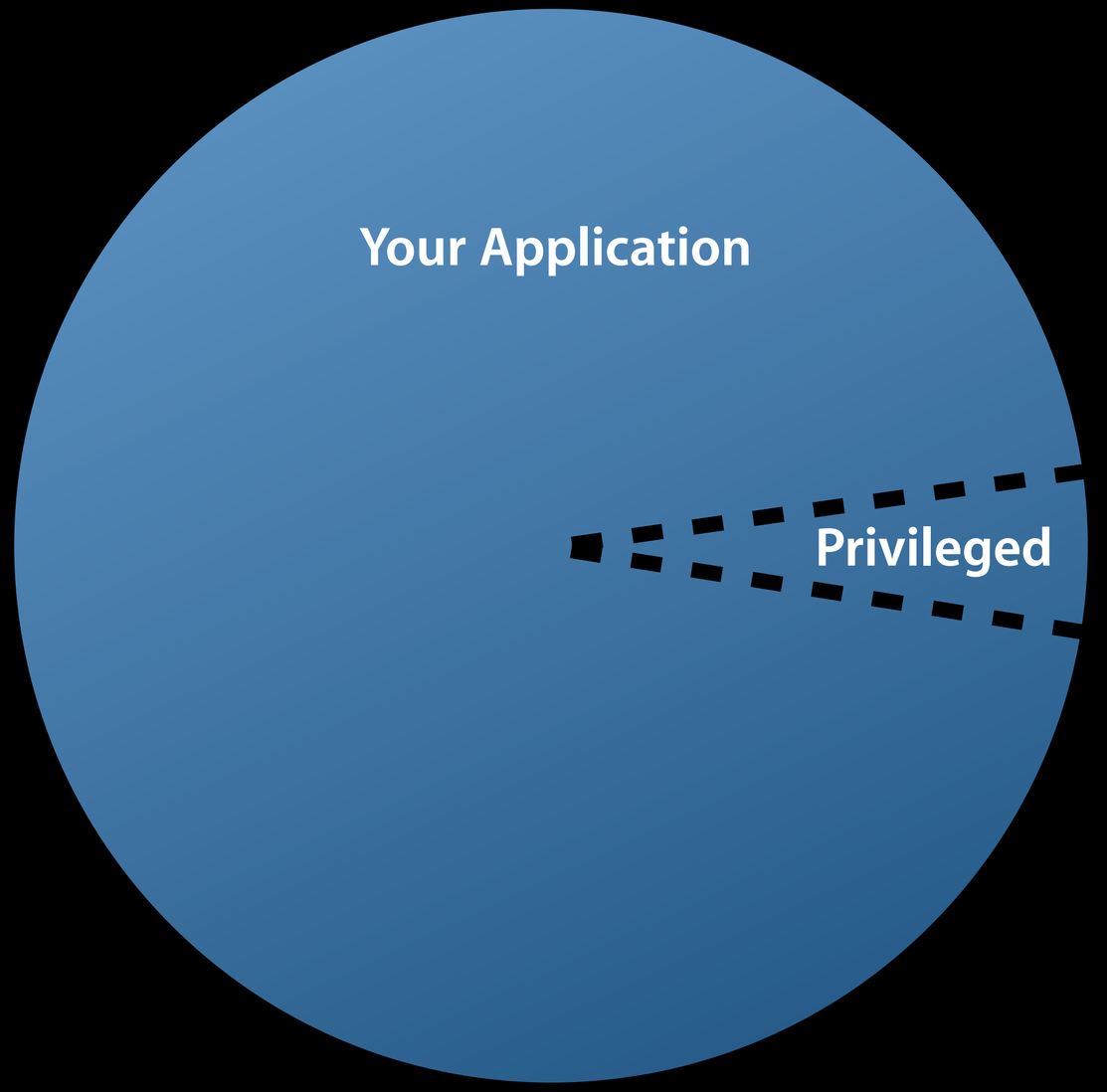
Least Privilege



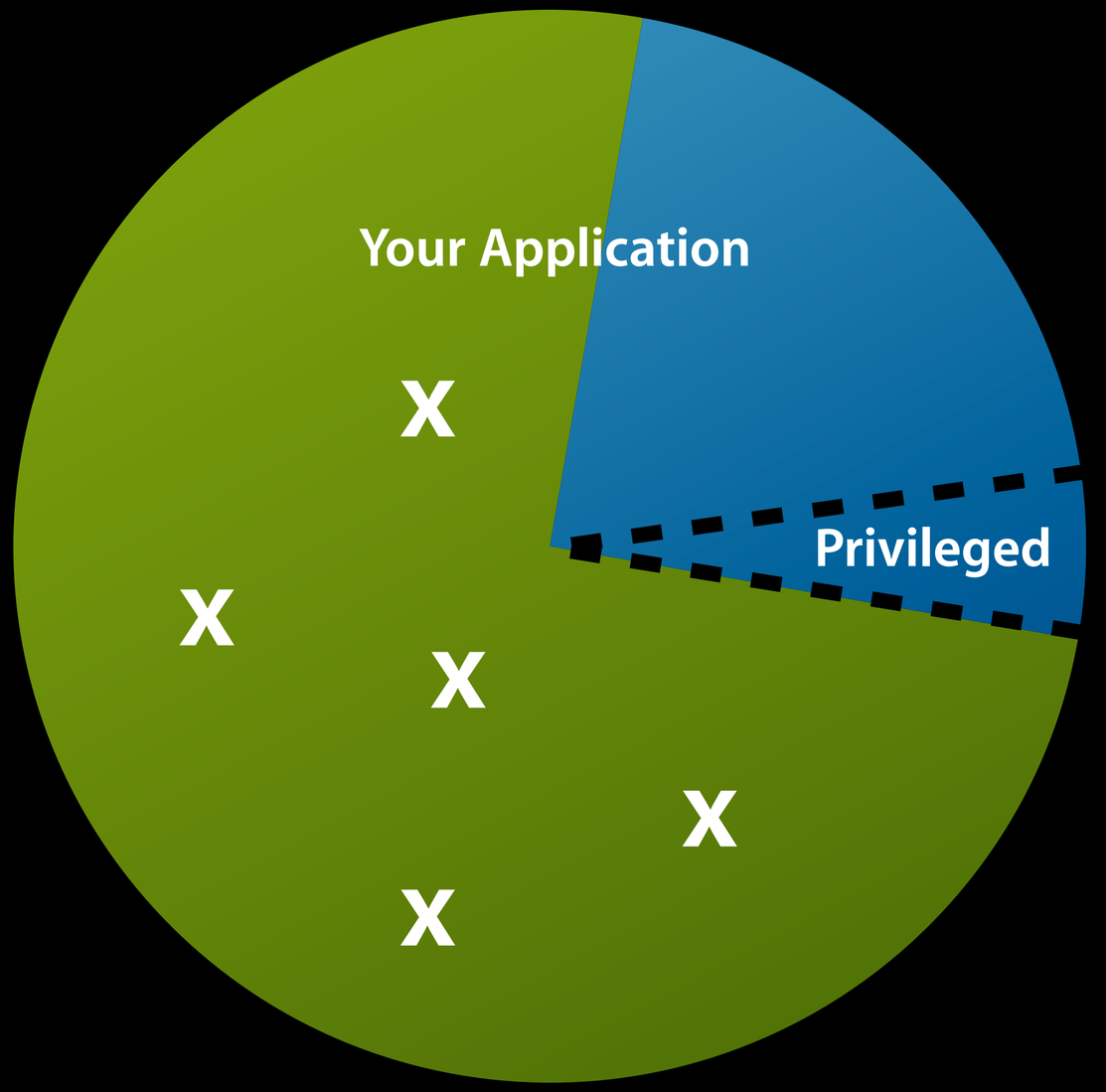
Least Privilege



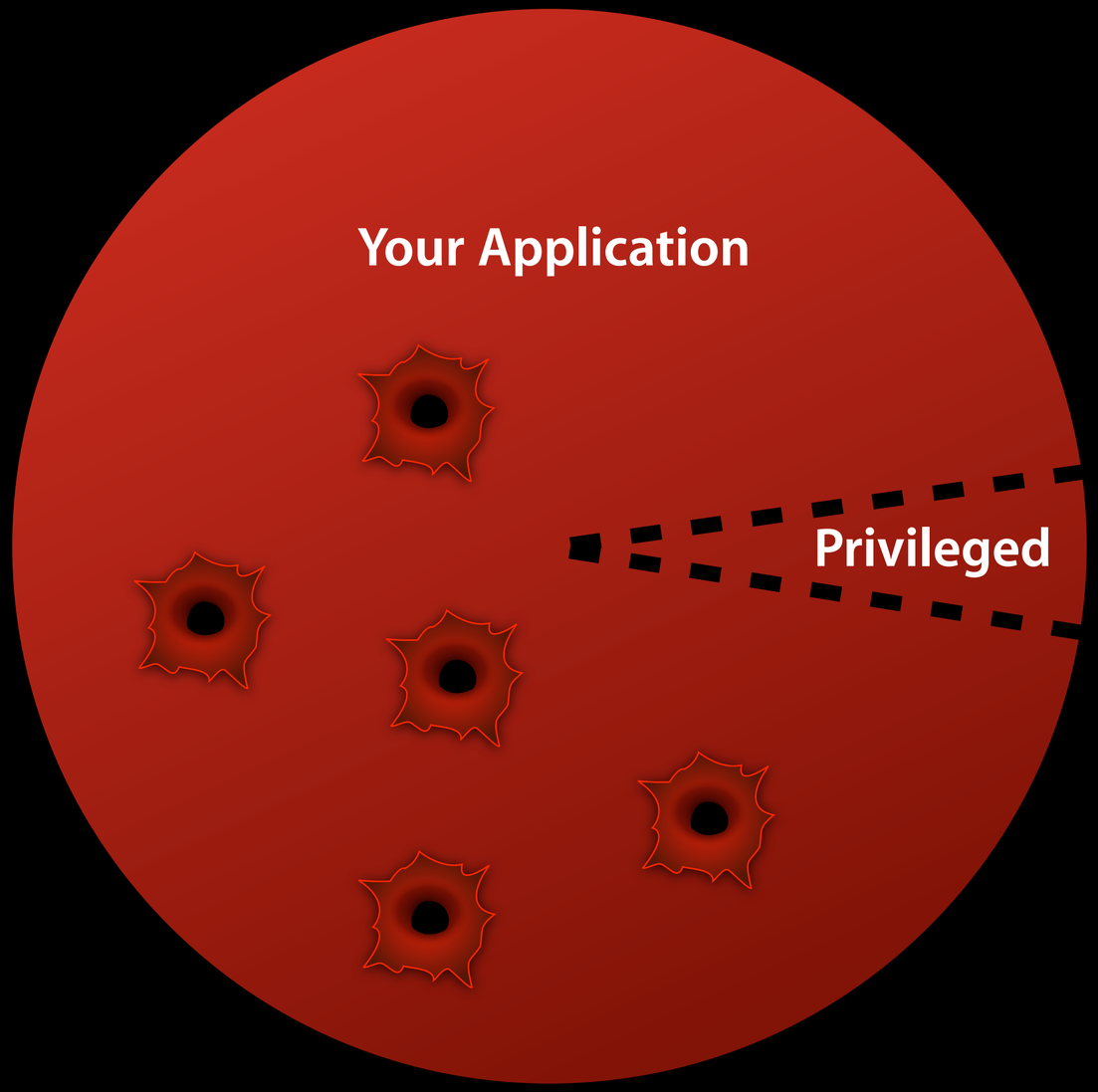
Least Privilege



Least Privilege



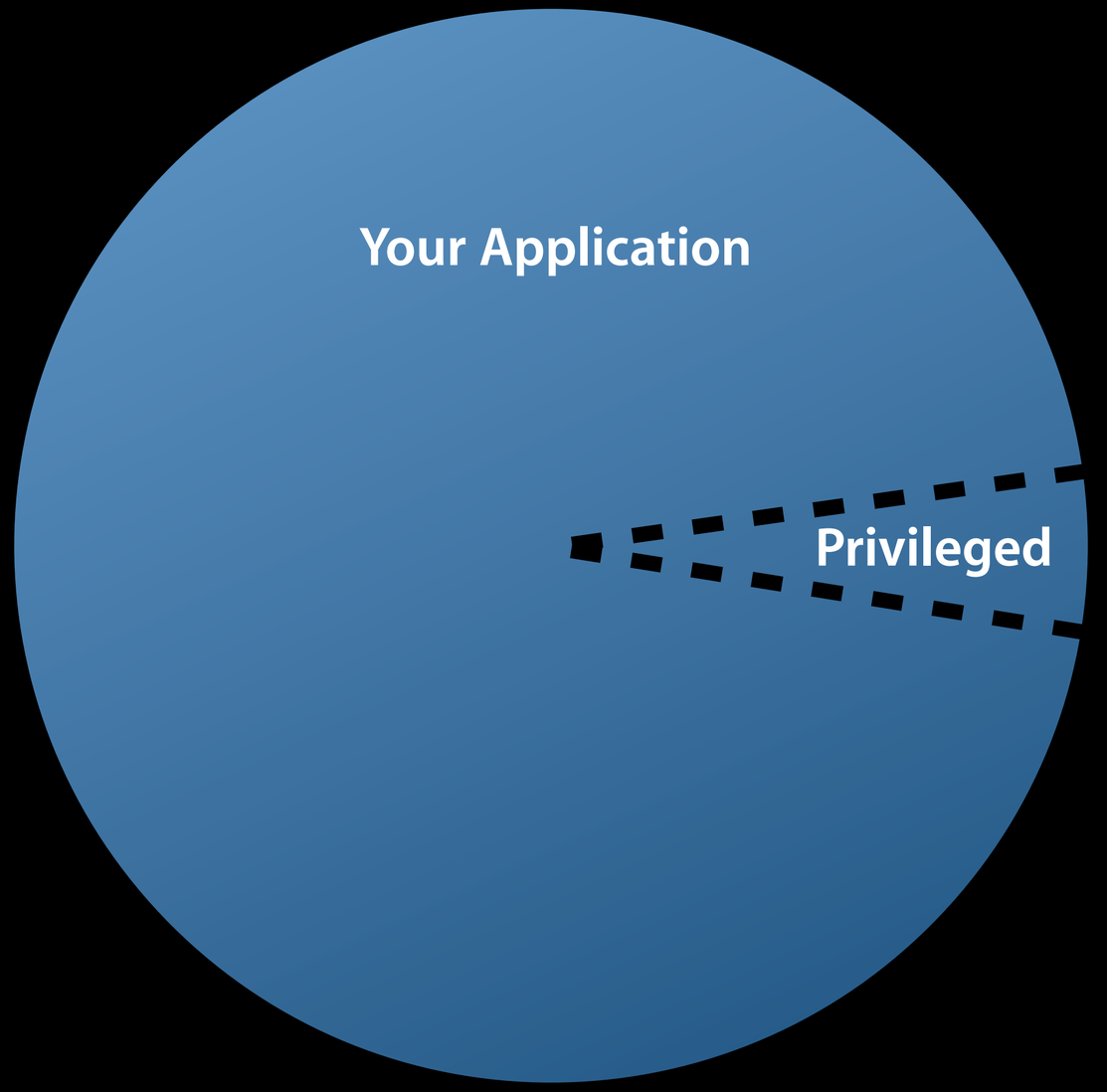
Least Privilege



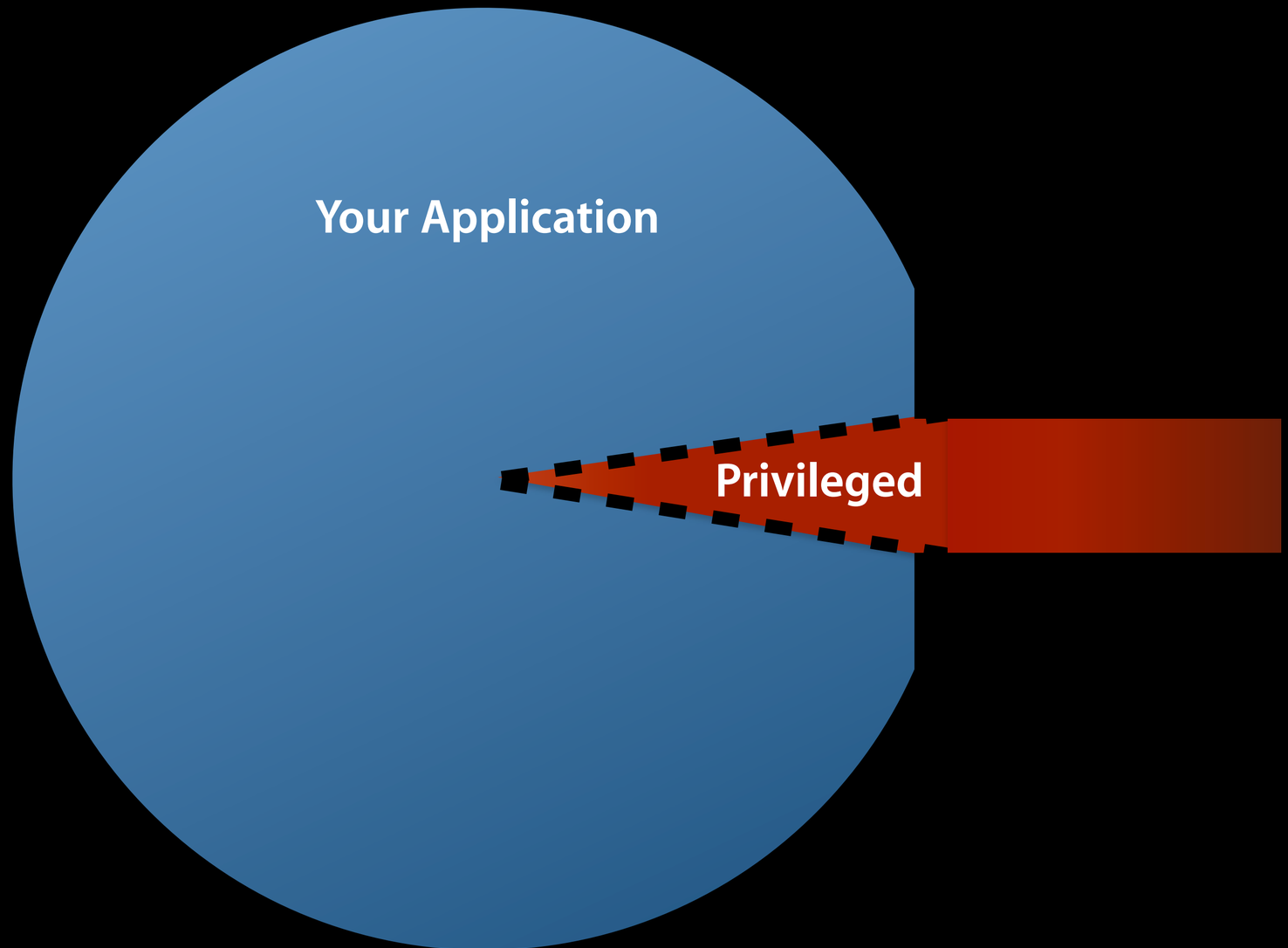
“Every program and every user of the system should operate using the least set of privileges necessary to complete the job.”

The Principal of Least Privilege
Saltzer, J.H. and Schroeder, M.D. ,
“The protection of information in computer systems.” (1975)

Least Privilege



Least Privilege



Overview

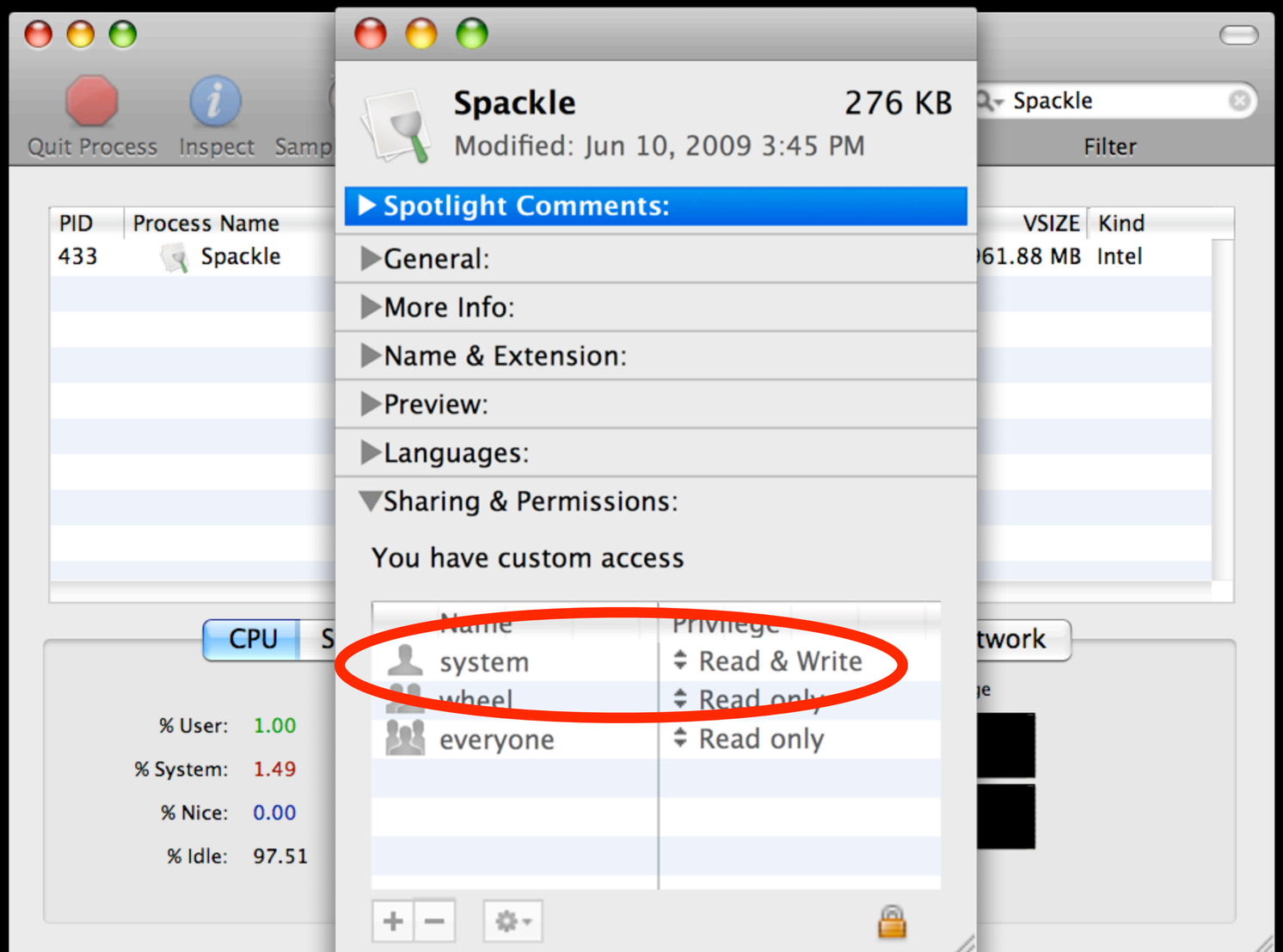
- Managing user privileges
  - Privileged helpers
  - Authorization Services
  - Code identity
  - Authorization Plugins

Privileged Helpers

Privileged Helpers

- Example application (“Spackle”)
- Self-updating feature
  - Check for a new version
  - Download a new version
  - Alert the user
  - Replace the existing version ⬅ Privileged operation
  - Relaunch the application

Privileged Helpers



Privileged Helpers

- Example application (“Spackle”)
- Self-updating feature
  - Check for a new version
  - Download a new version
  - Alert the user
  - Replace the existing version ⬅ Privileged operation
  - Relaunch the application

Privileged Helpers

Replace the existing version

Privileged Helpers

- Replace the existing version
  - Explain "sudo" to the user in a dialog
  - Set-user-ID application
  - AuthorizationExecuteWithPrivileges()
- Privileged helper tool

Privileged Helpers

- Goal: Privileges isolated
- Method
  - “helper” executable
  - Interprocess Communication (IPC)
- Only the helper tool needs privileges
- Only designated task

Privileged Helpers

- Spackle's helper
  - Has one sole task
  - Overwrites existing version
  - Protocol: Source, destination ➟ result

Spackle v1

Spackle v2

```
[[NSFileManager defaultManager] moveItemAtPath:downloaded
                                toPath:installed
                                error:&err];
```

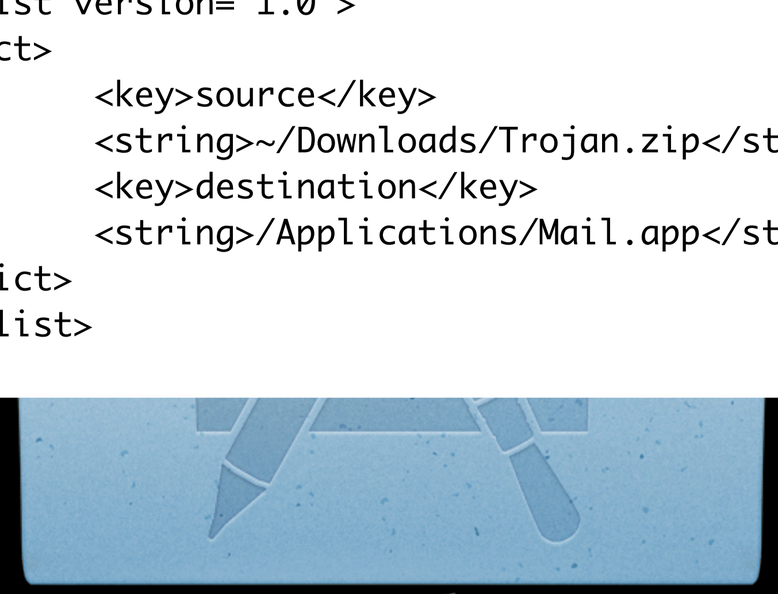
```
[[NSFileManager defaultManager] moveItemAtPath:downloaded
                                toPath:installed
                                error:&err];
```

Privileged Helpers

- launchd
- Register job for helper
- Launch on demand

Authorization Services

```
<plist version="1.0">
<dict>
        <key>source</key>
        <string>~/Downloads/Trojan.zip</string>
        <key>destination</key>
        <string>/Applications/Mail.app</string>
</dict>
</plist>
```



Spackle v1

helper

Grout

TrojanMail

Mail

```
[[NSFileManager defaultManager] moveItemAtPath:downloaded
                                toPath:installed
                                error:&err];
```

Authorization Services

- Authorization Services: A gatekeeper
- Separate policy and code
- User interaction required

Authorization Services

- Authorization Right request
- Policy evaluation
- Authorization result
- Right communicated to helper
- Validation

```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreate(NULL, NULL, kAuthorizationFlagDefaults, &auth);
AuthorizationItem rightItem[1] = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights     = { sizeof(rightItem)/sizeof(rightItem[0]), rightItem };
AuthorizationFlags authFlags   = kAuthorizationFlagInteractionAllowed |
                                 kAuthorizationFlagExtendRights;

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment, authFlags, NULL);
```

Spackle v1

helper

Spackle v2

Authorization Subsystem

```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreat
AuthorizationItem
AuthorizationRight
AuthorizationFlags

AuthorizationCopyR
```

```
0]), rightItem };

authFlags, NULL);
```

Spackle requires that you type your password.

Name: Administrator

Password: ••••••••••••••••••

Details

Right: com.example.Spackle.update

Application: Spackle

?

Cancel OK

Spackle v1

Spackle v2

helpe

Authorization Subsystem

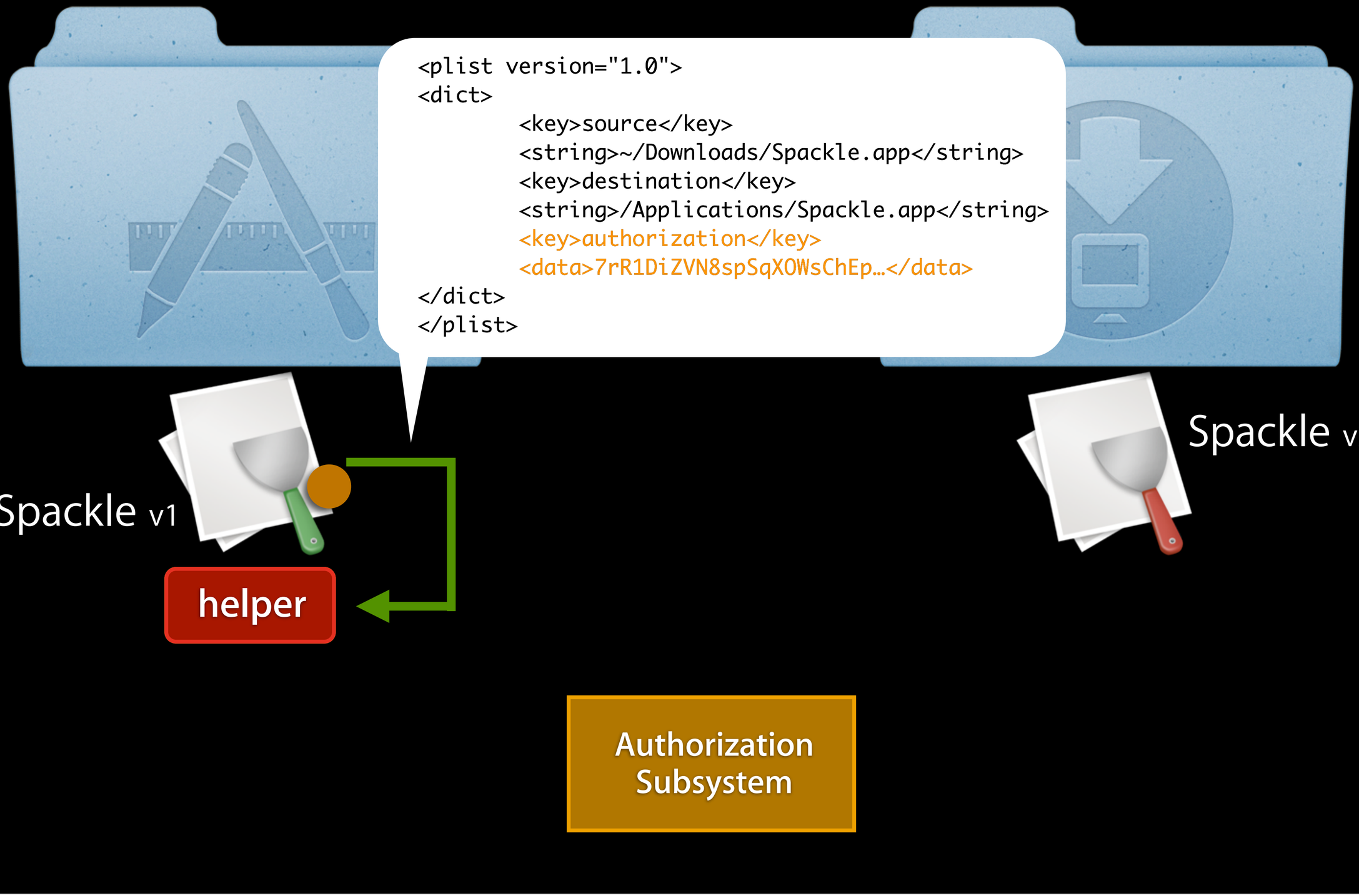
```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreate(NULL, NULL, kAuthorizationFlagDefaults, &auth);
AuthorizationItem rightItem[1] = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights     = { sizeof(rightItem)/sizeof(rightItem[0]), rightItem };
AuthorizationFlags authFlags   = kAuthorizationFlagInteractionAllowed |
                                 kAuthorizationFlagExtendRights;

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment, authFlags, NULL);
```

helper

Authorization Subsystem


<plist version="1.0">
<dict>
<key>source</key>
<string>~/Downloads/Spackle.app</string>
<key>destination</key>
<string>/Applications/Spackle.app</string>
<key>authorization</key>
<data>7rR1DiZVN8spSqXOWsChEp…</data>
</dict>
</plist>
Spackle v1
helper
Spackle v2
Authorization Subsystem

```
AuthorizationExternalForm   extAuth;

AuthorizationCreateFromExternalForm(&extAuth, &auth);

AuthorizationItem items[]  = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights = { sizeof(items[0])/sizeof(items), items };

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment,
                                       kAuthorizationFlagExtendRights,
                                       NULL);
```

Spackle v1

Spackle v2

helper

Authorization Subsystem

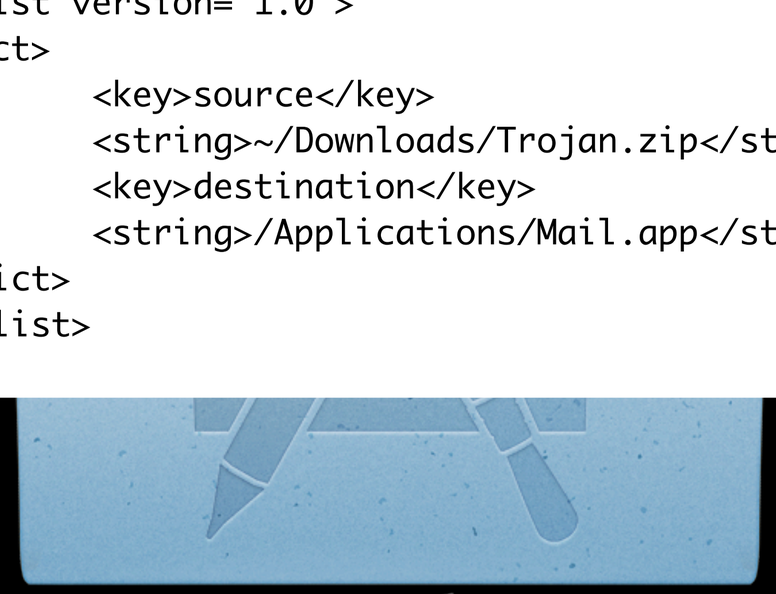
Spackle v1

helper

Spackle v2

```
[[NSFileManager defaultManager] moveItemAtPath:downloaded
                                toPath:installed
                                error:&err];
```

```
<plist version="1.0">
<dict>
        <key>source</key>
        <string>~/Downloads/Trojan.zip</string>
        <key>destination</key>
        <string>/Applications/Mail.app</string>
</dict>
</plist>
```



Spackle v1

helper

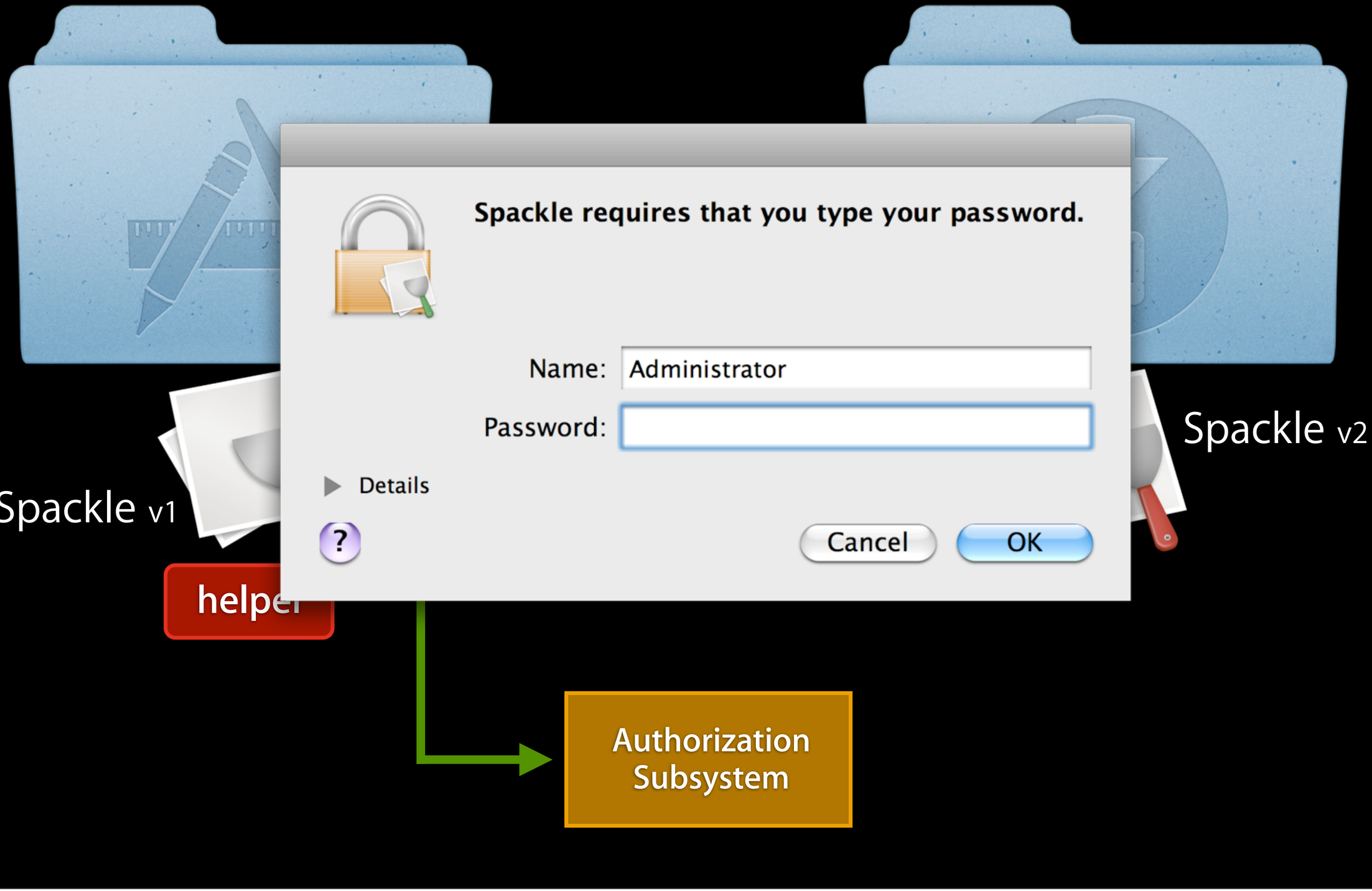
Grout

TrojanMail

Mail

Authorization Subsystem

Code Identity


Spackle requires that you type your password.
Name:
Administrator
Password:
Details
?
Cancel
OK
Spackle v1
Spackle v2
helper
Authorization Subsystem

Code Identity

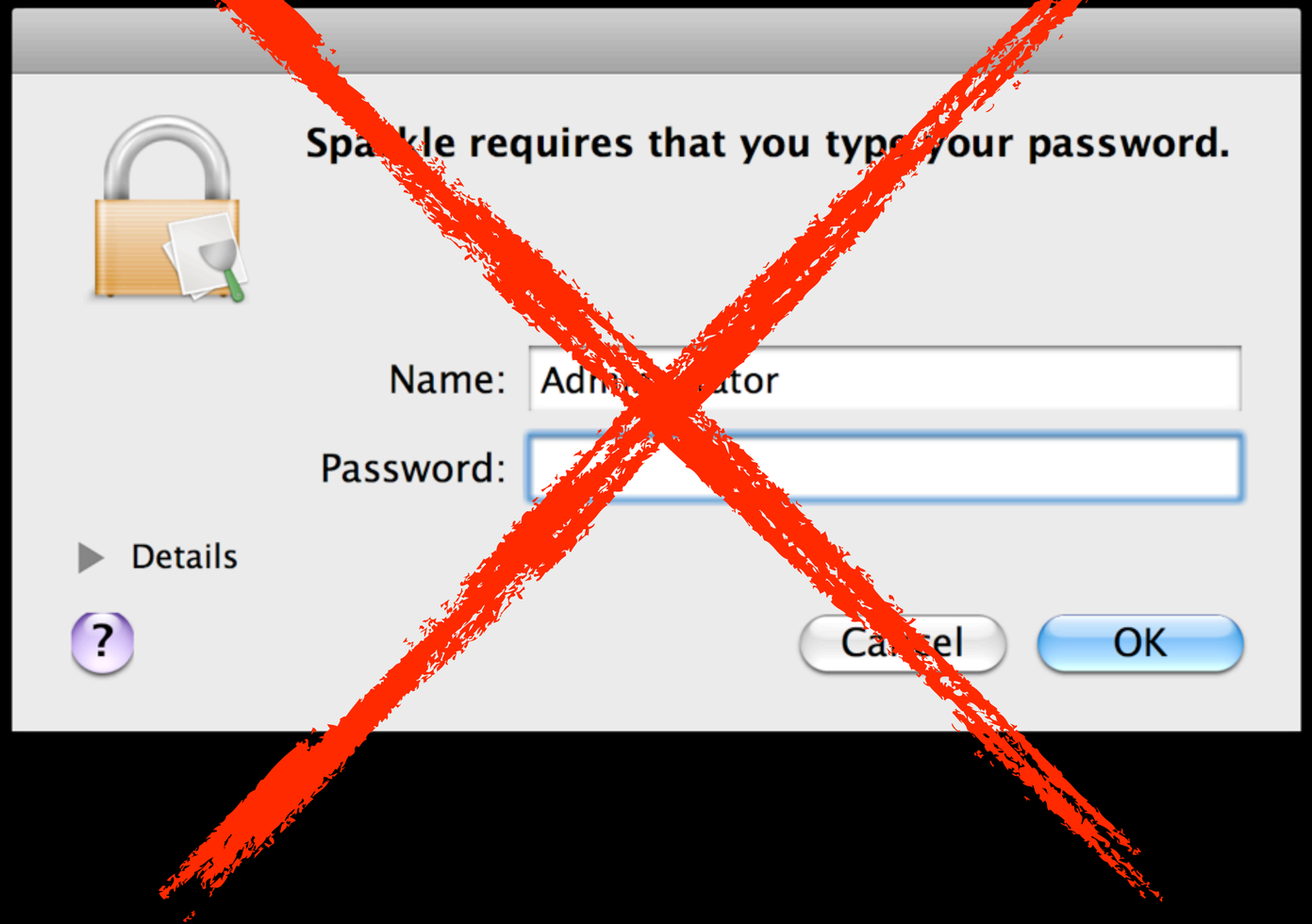
- Helper overwrites A with B
- Real goal: Update only
- Narrow privileges further

Code Identity

- Assign identity
- Helper validates identity
- Overwrite only if same app, newer version

Code Identity

- Now no arbitrary overwrite
- Possible to relax policy
- Skip admin authentication


Spa kle requires that you type your password.
Name:
Adm ator
Password:
Details
?
Ca el
OK

Authorization Plugins

Authorization Plugins

- Relaxed policy, but…
- Should require user interaction
- Helpful dialog
- Authorization Plugins

Authorization Plugins

- Plugin is installed
- Rules in policy database reference plugin
- Plugin invoked when Right requested

```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreate(NULL, NULL, kAuthorizationFlagDefaults, &auth);
AuthorizationItem rightItem[1] = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights     = { sizeof(rightItem[0])/sizeof(rightItem), rightItem };
AuthorizationFlags authFlags   = kAuthorizationFlagInteractionAllowed |
                                 kAuthorizationFlagExtendRights;

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment, authFlags, NULL);
```

helper

Authorization Subsystem

```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreate(NULL, NULL, kAuthorizationFlagDefaults, &auth);
Authoriz
Authoriz                                                        tItem };
Authoriz

Authoriz                                                        s, NULL);
```

Spackle

kle v2

Authorization Subsystem

Review

Technology	Use
launchd	Helper tool
Authorization Services	Access control
Code Identity	Safety, user experience
Authorization Plugins	Extensibility, user experience

Demo

Spackle v1

helper

Authorization Subsystem

Spackle v2

Spackle v1

helper

Spackle v2

Authorization Subsystem

Spackle v1

helper

Authorization Subsystem

Spackle v2

Spackle v1

helper

Spackle v2

A Little More Detail

- Authorization Exchange
- Policy Database
- Authorization Rights
- Authorization Rules
- Authorization Plugins

Authorization Exchange

Authorization Exchange - Step 1

- Spackle
  - AuthorizationCreate()
    - Create an AuthorizationRef
  - AuthorizationCopyRights()
    - Ask for the AuthorizationRight

Authorization Exchange - Step 2

- Spackle
  - AuthorizationMakeExternalForm()
    - Externalize the AuthorizationRef for the helper tool

```
const char *kSpackleHelperRight = "com.example.Spackle.update";

AuthorizationCreate(NULL, NULL, kAuthorizationFlagDefaults, &auth);
AuthorizationItem rightItem[1] = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights     = { sizeof(rightItem[0])/sizeof(rightItem), rightItem };
AuthorizationFlags authFlags   = kAuthorizationFlagInteractionAllowed |
                                 kAuthorizationFlagExtendRights;

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment, authFlags, NULL);
```

Spackle v1

helper

Spackle v2

Authorization Subsystem

```
AuthorizationExternalForm   extAuth;

AuthorizationMakeExternalForm(auth, &extAuth);
```

Spackle v1

helper

Spackle v2

Authorization Subsystem

Spackle v1

helper

Spackle v2

Authorization Subsystem

Authorization Exchange - Step 3

- helper
  - AuthorizationCreateFromExternalForm()
    - Create an AuthorizationRef
  - AuthorizationCopyRights()
    - Verify the AuthorizationRight

```
AuthorizationExternalForm   extAuth;

AuthorizationCreateFromExternalForm(&extAuth, &auth);

AuthorizationItem items[]  = { kSpackleHelperRight, 0, NULL, 0 };
AuthorizationRights rights = { sizeof(items[0])/sizeof(items), items };

AuthorizationCopyRights(auth, &rights, kAuthorizationEmptyEnvironment,
                                       kAuthorizationFlagExtendRights,
                                       NULL);
```

Spackle v1

helper

Spackle v2

Authorization Subsystem

Policy Database

Spackle v1

helper

Authorization Subsystem

com.example.Spackle.update

Spackle v2

Policy Database

- Rights and Rules
- Developers define Rights
- Administrators configure policy
- Currently an XML plist (/etc/authorization)
  - Database format subject to change

Policy Database

```
<plist version="1.0">
<dict>
        <key>comment</key>
        <string>The Authorization Policy Database</string>
        <key>rights</key>
        <dict>
                ...
        </dict>
        <key>rules</key>
        <dict>
                ...
        </dict>
</dict>
</plist>
```

Policy Database

- Installer Helper Tool
  - AuthorizationRightSet()
    - Add the Right to the policy database

```
const char *kSpackleHelperRight = "com.example.Spackle.update";


AuthorizationRightSet(NULL, kSpackleHelperRight,
                      CFSTR(kAuthorizationRuleIsAdmin),
                      CFSTR("Authorize updating Spackle"), NULL, NULL);
```

Spackle v1

helper

Policy Database

Authorization Subsystem

Installer Helper Tool

Spackle v2

Authorization Rights

Spackle v1

helper

Policy Database

Authorization Subsystem

com.example.Spackle.update

Spackle v2

Authorization Rights

- Rights in Policy Database
- Best practice: Right for each operation
- Right: Name and set of Rules

Authorization Rights

```
<key>com.MyCompany.MyApplication.WhatThisRightDoes</key>
<dict>
        <key>authenticate-user</key>
        <true/>
        <key>session-owner</key>
        <true/>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <false/>
        <key>timeout</key>
        <integer>0</integer>
</dict>
```

Authorization Rights

- Naming convention
- Example: com.MyCompany.MyApplication. WhatThisRightDoes
- Wildcard
  - A right ending in '.'
    - Implies a set of identical rights
- Reserved names
  - The "system." prefix
  - The "config." prefix

Authorization Rules

Spackle v1

helper

Policy Database

Authorization Subsystem

com.example.Spackle.update

Spackle v2

Authorization Rules

- Rules in Policy Database
- Rights reference Rules to be satisfied
- Properties
- Classes

Authorization Rules

- Properties
  - Authenticate-user
  - Timeout
  - Shared

Authorization Rights

```
<key>com.MyCompany.MyApplication.WhatThisRightDoes</key>
<dict>
        <key>authenticate-user</key>
        <true/>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <false/>
        <key>timeout</key>
        <integer>0</integer>
</dict>
```

Authorization Rights

```
<key>com.MyCompany.MyApplication.WhatThisRightDoes</key>
<dict>
        <key>authenticate-user</key>
        <true/>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <false/>
        <key>timeout</key>
        <integer>0</integer>
</dict>
```

Authorization Rights

```
<key>com.MyCompany.MyApplication.WhatThisRightDoes</key>
<dict>
        <key>authenticate-user</key>
        <true/>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <false/>
        <key>timeout</key>
        <integer>0</integer>
</dict>
```

Authorization Rules

- Classes
  - Allow
  - Deny
  - User
  - Evaluate-mechanisms
  - Rule (delegation)
  - Default rule

Authorization Rules

- Classes
  - User
    - User-in-group
    - Session owner

Authorization Rules

```
<key>com.apple.activitymonitor.kill</key>
<dict>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <false/>
        <key>timeout</key>
        <integer>0</integer>
</dict>
```

Authorization Rules

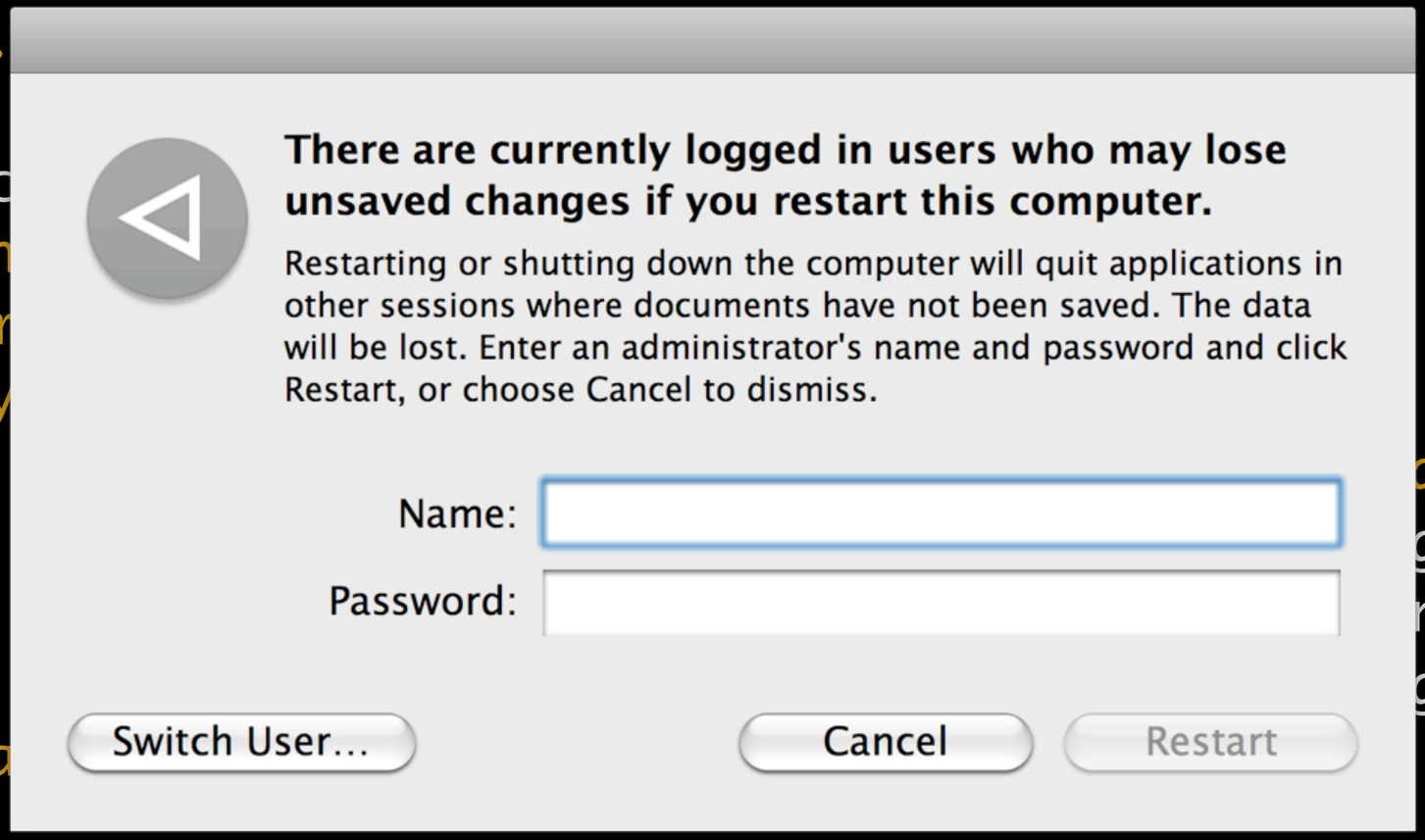
```
<key>system.identity.write.self</key>
<dict>
        <key>authenticate-user</key>
        <false/>
        <key>class</key>
        <string>user</string>
        <key>session-owner</key>
        <true/>
</dict>
```

Authorization Rules

- Classes
  - Evaluate-mechanisms
    - Plugins each provide one or more Mechanisms
    - Mechanisms acquire or operate on authorization data
    - Run as root, or as the Security Agent role account

Authorization Rules

```
<key>system.
<dict>
        <key>c
        <strin
        <key>m
        <array
                                                    d</string>
                                                    g>
                                                    ring>
                                                    g>
        </arra
</dict>
```



Authorization Rules

- Classes
  - Rule (delegation)
    - Logical AND (n-of-n)
    - Subset (k-of-n)

Authorization Rules

```
<key>com.apple.DiskManagement.</key>
<dict>
        <key>class</key>
        <string>rule</string>
        <key>k-of-n</key>
        <integer>1</integer>
        <key>rule</key>
        <array>
                <string>is-root</string>
                <string>is-admin</string>
                <string>default</string>
        </array>
        <key>shared</key>
        <true/>
</dict>
```

Authorization Rules

- Classes
  - Default rule
    - Used if no rule is specified for a right
    - Group admin
    - Allow user to authenticate, if needed
    - 5 minute timeout
    - Shared

Authorization Rules

```
<key>system.</key>
<dict>
        <key>rule</key>
        <string>default</string>
</dict>
```

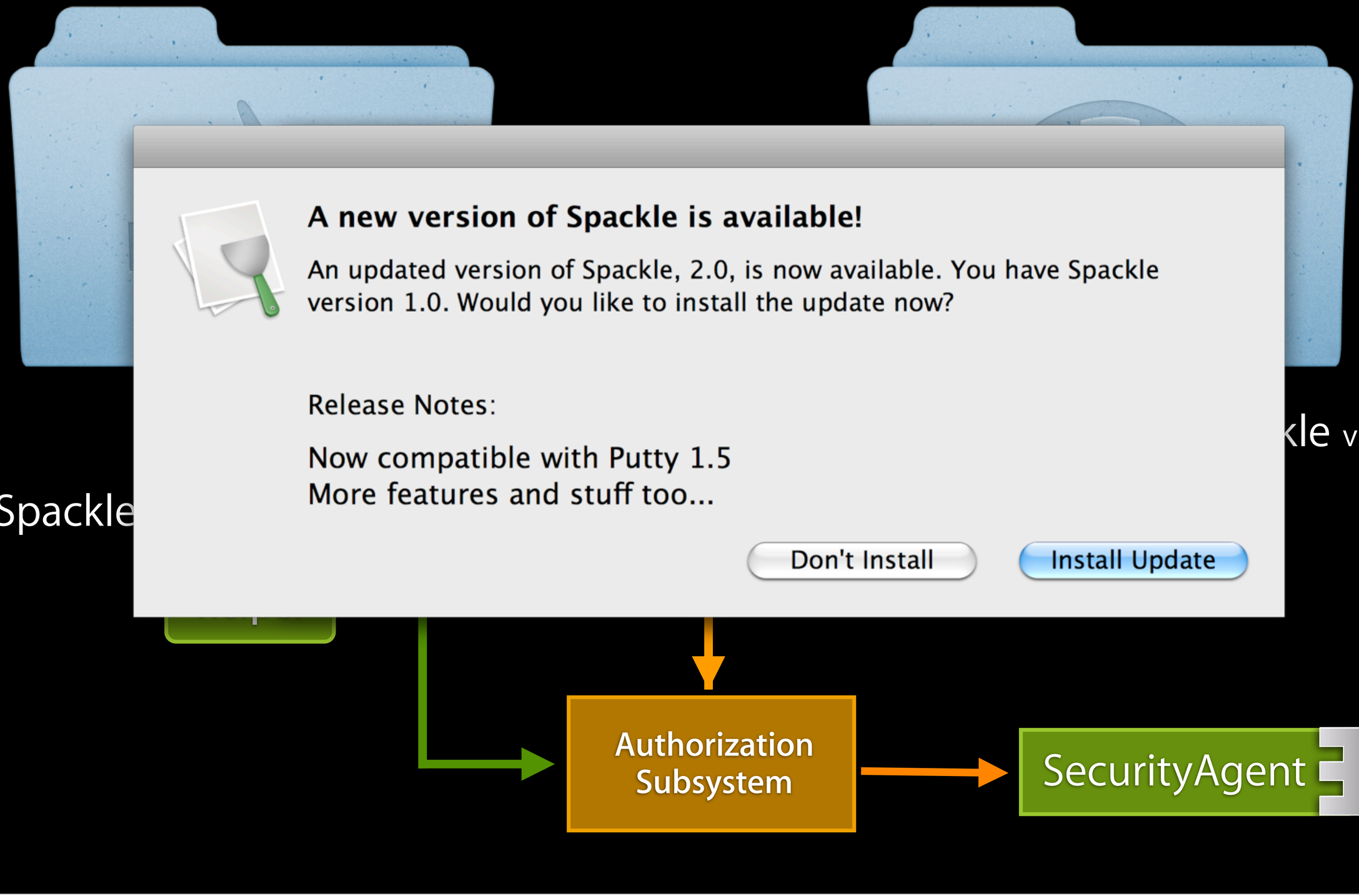
Authorization Rules

```
<key>system.</key>
<dict>
        <key>rule</key>
        <string>default</string>
</dict>
```

Authorization Rules

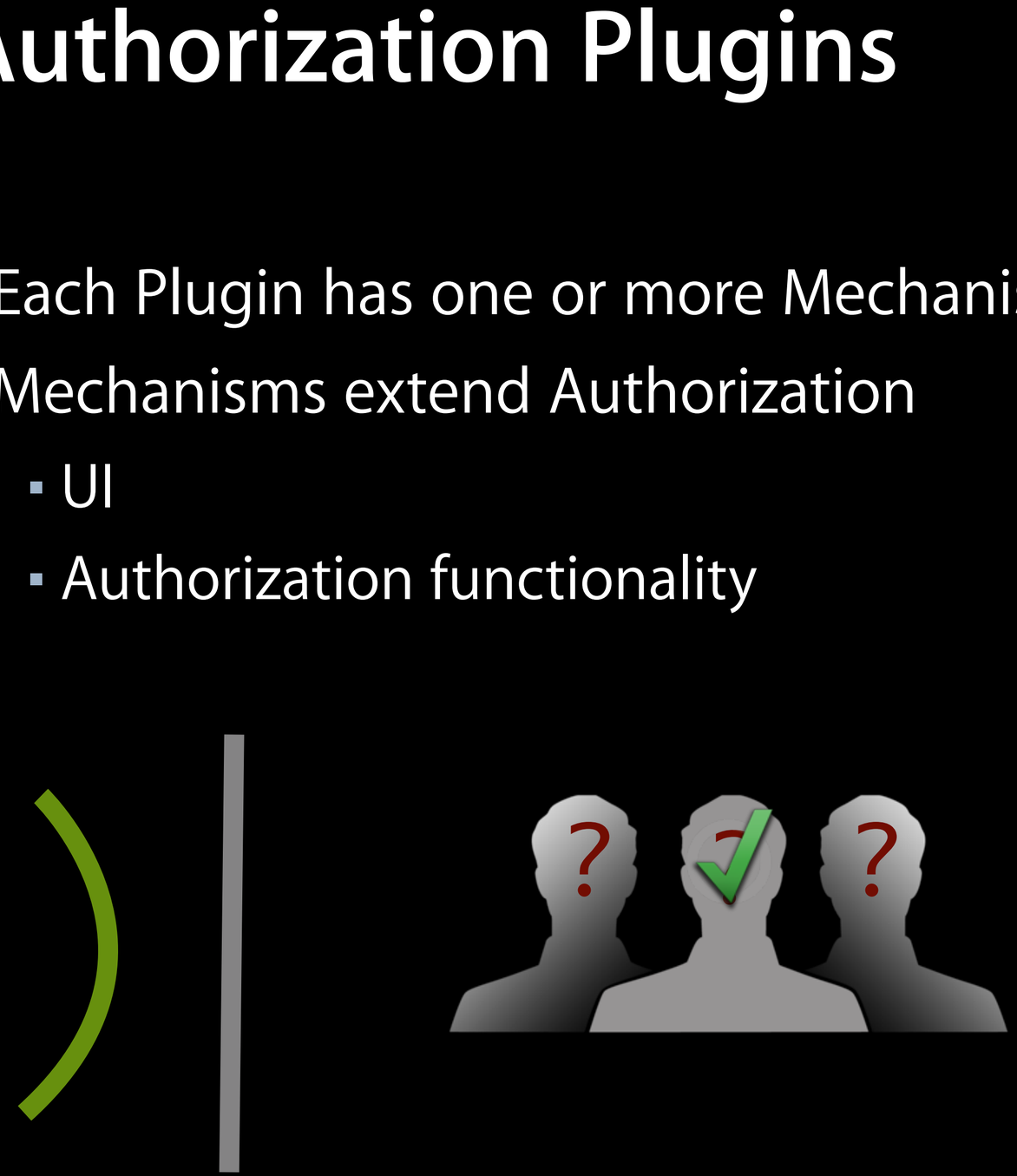
```
<key>default</key>
<dict>
        <key>class</key>
        <string>user</string>
        <key>group</key>
        <string>admin</string>
        <key>shared</key>
        <true/>
        <key>timeout</key>
        <integer>300</integer>
</dict>
```

Authorization Plugins

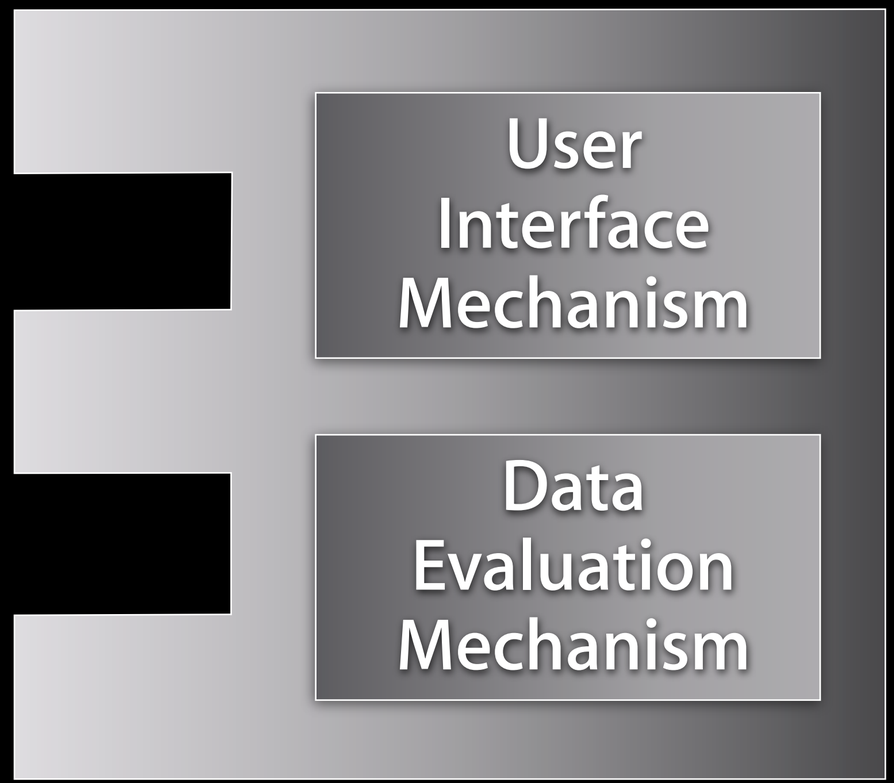

A new version of Spackle is available!
An updated version of Spackle, 2.0, is now available. You have Spackle version 1.0. Would you like to install the update now?
Release Notes:
Now compatible with Putty 1.5
More features and stuff too...
Don't Install
Install Update
Spackle
kle v2
Authorization Subsystem
SecurityAgent

Authorization Plugins

- Each Plugin has one or more Mechanisms
- Mechanisms extend Authorization
  - UI
  - Authorization functionality



Example Plugin



Authorization Plugins

- Location
- Lifecycle
- UI

Authorization Plugins

- Location
  - /Library/Security/SecurityAgentPlugins/*PluginName*.bundle
  - not ~/Library

Authorization Plugins

- Lifecycle
  - Plugin Instantiation
    - Mechanism Creation
    - Mechanism Invocation
    - Mechanism Deactivation
    - Mechanism Destruction
  - Plugin Cleanup

Authorization Plugins

- UI
  - SFAuthorizationPluginView
    - Base for custom user interface
    - Adds view to authentication dialogs
    - Objective-C based

Spackle v1

Spackle v2

Spackle v1

Spackle v2

Authorization Subsystem

SecurityAgent

Spackle v1

helper

Spackle v2

Authorization Subsystem

Spackle v1

helper

Spackle v2

Authorization Subsystem

Spackle v1

helper

Spackle v2

Authorization Subsystem

More Information

Craig Keithley
Security Technology Evangelist
keithley@apple.com

Sample code
BetterAuthorizationSample
http://developer.apple.com/samplecode/BetterAuthorizationSample/

Sample code
Authorization Plugins
http://developer.apple.com/samplecode/NullAuthPlugin/
http://developer.apple.com/samplecode/NameAndPassword/

More Information

Header Files

Authorization Services
<Security/Authorization.h>
<Security/AuthorizationDB.h>
<Security/AuthorizationPlugin.h>
<SecurityInterface/SFAuthorizationPluginView.h>

More Information

Documentation
Performing Privileged Operations With Authorization Services
http://developer.apple.com/documentation/Security/Conceptual/authorization_concepts/

Documentation
Technical Note TN2038, “Daemons and Agents” (launchd agents)
http://developer.apple.com/technotes/tn2005/th2083.html

Related Sessions

Assigning Your Application an Identity with Code Signing	Nob Hill Tuesday 9:00AM
Managing Processes with launchd	Russian Hill Friday 2:00PM

Summary

- Factor out privileged code into a helper tool
- Use launchd for helper tools
- Use Authorization Services to gate use of the helper tool
- Implement code identity to eliminate redundant dialogs
- Customize user experience and extend Authorization with Plugins

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09